



C1 - 12 循环嵌套

// 学习目标

- 掌握 for - for 循环嵌套
- 掌握 for - while 循环嵌套
- 掌握 break 和 continue 的用法

// for - for 循环嵌套

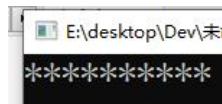


还记得如何输出满天星吗？

```
*****!
*****!
*****!
*****!
*****!
*****!
```

至少打印一行星星对我们来说没有难度：

```
#include <iostream>
using namespace std ;
int main(){
    for ( int i=1 ; i <= 10 ; i++) { // 循环 10 次
        cout << "*" ; // 每次输出一颗星星
    }
    return 0 ;
}
```



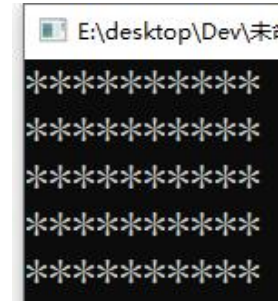
既然这样可以输出 1 行星星，
那如果输出 5 行呢？写 5 个这
样的循环吗？

不要，太麻烦了



如果把刚刚使用的循环放在另一个需要循环 5 次的循环里会怎样呢？

```
#include <iostream>
using namespace std ;
int main(){
    for ( int k=1 ; k <= 5 ; k++){ // 循环 5 次
        for ( int i=1 ; i <= 10 ; i++){ // 循环 10 次
            cout << "*" ; // 每次输出一颗星星
        }
        cout << endl ;
    }
    return 0 ;
}
```



// for - while 循环嵌套

之前我们在做“数位分离”时就是使用了 while 循环，但是是给 1 个数字做数位分离，如果把 1~n 之间的数字都拿出来做一次数位分离怎么实现呢？

```
#include <iostream>
using namespace std ;
int main(){
    int x = 1234 , y = 0 ;
    while ( x ){
        y = y * 10 + x % 10 ;
        x /= 10 ;
    }
    cout << y ;
    return 0 ;
}
```

1 个数的数位分离

```
#include <iostream>
using namespace std ;
int main(){
    int n ;
    cin >> n ;
    for ( int i=1 ; i <= n ; i++){ // n 次循环，由 i 表示 1~n 之间的数字
        int y = 0 ; // 存储每一个数字进行数位分离之后的结果！
        int x = i ; // 由 x 替代 i 进行数位分离，避免打破循环规律
        while ( x ){ // 内循环，进行数位分离
            y = y * 10 + x % 10 ;
            x /= 10 ;
        }
        cout << y ;
    }
    return 0 ;
}
```

n 个数的数位分离

// break 和 continue



结束所在的循环语句

```
int main(){  
    for(int i = 1;i <= 5;i++){  
        cout << 1;  
        break;  
        cout << 2;  
    }  
    return 0;  
}
```

1



结束本次循环，继续下一次

```
int main(){  
    for(int i = 1;i <= 5;i++){  
        cout << 1;  
        continue;  
        cout << 2;  
    }  
    return 0;  
}
```

11111